

Approximate nearest neighbor search for ℓ_p -spaces ($2 < p < \infty$) via embeddings

Yair Bartal ^{*} Lee-Ad Gottlieb [†]

December 8, 2015

Abstract

While the problem of approximate nearest neighbor search has been well-studied for Euclidean space and ℓ_1 , few non-trivial algorithms are known for ℓ_p when $2 < p < \infty$. In this paper, we revisit this fundamental problem and present approximate nearest-neighbor search algorithms which give the first non-trivial approximation factor guarantees in this setting.

1 Introduction

Nearest neighbor search (NNS) is one of the basic operations computed on data sets comprising numeric vectors, i.e. points. The problem asks to preprocess a d -dimensional set V of $n = |V|$ vectors residing in a certain space M , so that given a new query point $q \in M$, a point nearest to q in V can be located efficiently. This problem has applications in data mining, database queries and related fields.

When the ambient space M is a high-dimensional ℓ_p -space,¹ NNS may require significant computation time, and this is due to the inherent complexity of the metric. For example, for Euclidean vectors (ℓ_2 -space), Clarkson [13] gave an $O(n^{\lceil d/2 \rceil (1+\varepsilon)})$ size data structure that answers exact NNS queries in $O(\log n)$ time (with constant factors in the bounds depending on constant $\varepsilon > 0$), and claimed that the exponential dependence on d is a manifestation of Bellman’s [7] “curse of dimensionality.” This has led researchers to consider the c -approximate nearest neighbor problem (ANN), where the goal is to find a point in V whose distance to q is within a factor c of the distance to q ’s true nearest neighbor in V . In the Euclidean setting, celebrated results of Kushilevitz, Ostrovsky and Rabani [32] (see also [36, 37]) and Indyk and Motwani [26, 21] achieved polynomial-size data structures which return a $(1 + \varepsilon)$ -ANN in query time polynomial in $d \log n$ (when $\varepsilon > 0$ is any constant). These results can be extended to all ℓ_p with $1 \leq p \leq 2$.

However, the more difficult regime of $p > 2$ is significantly less well understood. Recalling that for any vector v and $p > 2$, $d^{\frac{1}{p}-\frac{1}{2}} \|v\|_2 \leq \|v\|_p \leq \|v\|_2$, we conclude that simply running an ℓ_2 ANN algorithm on $V \subset \ell_p$ (that is, treating V as if it resided in ℓ_2) will return an $O(d^{\frac{1}{2}-\frac{1}{p}})$ -ANN in polylog

^{*}Hebrew University. Work supported in part by an Israel Science Foundation grant #1609/11. Email: yair@cs.huji.ac.il

[†]Ariel University. Work supported in part by an Israel Science Foundation grant #755/15. Email: leead@ariel.ac.il

¹This is a space equipped with a Minkowski norm, which defines the distance between two d -dimensional vectors x, y as $\|x - y\|_p = (\sum_{i=1}^d |x_i - y_i|^p)^{1/p}$.

time. If we allow exponential space, then a $(1 + \varepsilon)$ -ANN can be found in query time $O(d \log n)$ and space $\varepsilon^{-O(d)}n$ by utilizing an approximate Voronoi diagram [21, 4]. (These structures were developed for Euclidean spaces, but apply to all ℓ_p , $p \geq 1$ as well [23].) For ℓ_∞ , Indyk [24] gave a polynomial-size structure which answers $O(\log \log d)$ -approximate queries in $d \log^{O(1)} n$ query time, and remarkably there are indications that this bound may be optimal [3]. Since for any vector v we have $\|v\|_\infty \leq \|v\|_p \leq d^{1/p} \|v\|_\infty$, Indyk's ℓ_∞ structure gives a $O(d^{1/p} \log \log d)$ -ANN algorithm for all $p > 2$, and in particular an $O(\log \log d)$ -ANN whenever $p = \Omega(\log d)$.

Our contribution. We revisit the problem of ANN in ℓ_p spaces for $2 < p < \infty$, and give improvements over what was previously known. Note that the ℓ_p -norm for this regime finds application in fields such as image resolution [20, 19], time-series comparison [39], and k -means clustering [16]. We are interested in polynomial-size structures that have query time polynomial in $d \log n$. Hence we shall make the simplifying assumptions that $d = \omega(\log n)$ and $d = n^{o(1)}$: If $d = O(\log n)$ then approximate Voronoi diagrams may be used, and if $d = n^{\Omega(1)}$ then comparing the query point q to each point in V in a brute-force manner can be done in time $O(dn) = d^{O(1)}$.²

Our first result is an extension of Indyk's ℓ_∞ algorithm to smaller values of p . Exploiting the max-stability property of Fréchet random variables, we give in Section 3.1 a randomized embedding from ℓ_p into ℓ_∞ that is non-contractive and has expansion $O(\log^{1/p} n)$. This means that points distant from query point q remain far, while the distance from q to its nearest neighbor increases by at most a factor of $O(\log^{1/p} n)$. We then run Indyk's algorithm on the embedded space, and the result must be a $O(\log \log d \log^{1/p} n)$ -ANN in the origin space. We refine this technique in Section 3.2 to obtain better bounds for spaces with low doubling dimension.

Having extended the range in which Indyk's ℓ_∞ algorithm is applicable, we proceed to introduce an embedding which greatly extends the range for which the ℓ_2 algorithms are applicable. In Section 4, we introduce the Mazur map as an algorithmic tool. This mapping allows us to embed ℓ_p into ℓ_2 , and we then solve ANN in the embedded space. Although the Mazur map induces distortion dependent on the diameter of the set, thereby confounding the ANN search, we show that the mapping can be applied to small low-diameter subproblems. Our final result is a polynomial-size structure which answers $2^{O(p)}$ -approximate queries in time polynomial in $d \log n$ (Theorem 4.4). This yields non-trivial results for $p = o(\log d)$. Comparing this result with the one above:

- When $p = O(\sqrt{\log \log n})$, the $2^{O(p)}$ -ANN algorithm is best.
- When $p = \Omega(\sqrt{\log \log n})$ the $O(\log \log d \log^{1/p} n)$ -ANN algorithm is best.

Note that the worst case is when $p = \Theta(\sqrt{\log \log n})$, where the approximation ratio is $2^{O(p)} = 2^{O(\sqrt{\log \log n})}$. As we have assumed $d = \omega(\log n)$, the above bound is $2^{o(\sqrt{\log d})}$, much better than the previously known $d^{O(1)}$ -factor approximations.

1.1 Related work

For Euclidean space, Chan [11] gave a deterministic construction which gives an $O(d^{3/2})$ -ANN, in time $O(d^2 \log n)$ and using polynomial space (see also [8]). For ℓ_p , Neylon [35] gave an $O(d)$ -ANN structure which runs in $O(d^2 \log n)$ time and uses $\tilde{O}(dn)$ space.

²We recall also that there exists an oblivious mapping for all ℓ_p that embeds ℓ_p^m into ℓ_p^d for $d = \binom{n}{2}$ dimensions [18, 5].

Stoev *et al.* [38] and Andoni [2] both used random variables with max-stability or min-stability to estimate the p -th moment of a vector, or of the difference between two vectors. Faragó [17] presented an elegant oblivious embedding from ℓ_p^d to $\ell_\infty^{2^{O(d)}}$ with arbitrarily low distortion, and the existence of an embedding with these properties had been alluded to in Indyk’s survey [25]. We observe that one may utilize Farago’s embedding to map ℓ_p into ℓ_∞ and then compute the nearest neighbor in the embedded space using Indyk’s ℓ_∞ structure. This in fact yields an $O(\log d)$ -ANN, but in time and space exponential in d , so approximate Voronoi diagrams are better for this problem.

For ANN in general metric spaces, Krauthgamer and Lee [31] showed that the doubling dimension can be used to control the search runtime: For a metric point set S , they constructed a polynomial-size structure which finds an $O(1)$ -ANN in time $2^{O(\text{ddim}(S))} \log \Delta$, where $\Delta = \Delta(S)$ is the *aspect ratio* of S , the ratio between the maximum and minimum inter-point distances in S . The space requirements of this data structure were later improved by Beygelzimer *et al.* [9]. Har-Peled and Mendel [22] and Cole and Gottlieb [15] showed how to replace the dependence on $\log \Delta$ with dependence on $\log n$. Other related research on nearest neighbor searches have focused on various assumptions concerning the metric space. Clarkson [14] made assumptions concerning the probability distribution from which the database and query points are drawn, and developed two randomized data structures for exact nearest neighbor. Karger and Ruhl [30] introduced the notion of growth-constrained metrics (elsewhere called the KR-dimension) which is a weaker notion than that of the doubling dimension. A survey of proximity searches in metric space appeared in [12].

Subsequent to the dissemination of the results in this paper, we were advised of a manuscript of Naor and Rabani [34] (mentioned in [33, Remark 4.2]) which gives a similar $2^{O(p)}$ -ANN algorithm for $p > 2$, also utilizing the Mazur map. In personal communication, Assaf Naor broached the question of better dependence on p in the $2^{O(p)}$ approximation bound of Theorem 4.4. He noted that all *uniform* embeddings of ℓ_p ($p > 2$) into ℓ_2 (such as the Mazur map) possess distortion exponential in p [33, Lemma 5.2]. Non-uniform embeddings of ℓ_p into ℓ_2 may possess better distortion bounds.

1.2 Preliminaries

Embeddings and metric transforms. A much celebrated result for dimension reduction is the well-known l_2 flattening lemma of Johnson and Lindenstrauss [28]: For every n -point subset of l_2 and every $0 < \varepsilon < 1$, there exists a mapping into l_2^k that preserves all inter-point distances in the set within factor $1 + \varepsilon$, with target dimension $k = O(\varepsilon^{-2} \log n)$.

Following Batu *et al.* [6], we define an *oblivious* embedding to be an embedding which can be computed for any point of a database or query set, without knowledge of any other point in these sets. (This differs slightly from the definition put forth by Indyk and Naor [27].) Familiar oblivious embeddings include standard implementations of the Johnson-Lindenstrauss transform for l_2 [28], the dimension reduction mapping of Ostrovsky and Rabani [37] for the Hamming cube, and the embedding of Johnson and Schechtman [29] for ℓ_p , $p \leq 2$.

An embedding of X into Y with *distortion* D is a mapping $f : X \rightarrow Y$ such that for all $x, y \in X$,

$$1 \leq c \cdot \frac{d_Y(f(x), f(y))}{d_X(x, y)} \leq D,$$

where c is any scaling constant. An embedding is *non-contractive* if for all $x, y \in X$,

$$\frac{d_Y(f(x), f(y))}{d_X(x, y)} \geq 1.$$

It is *non-expansive* (or *Lipschitz*) if for all $x, y \in X$,

$$\frac{d_Y(f(x), f(y))}{d_X(x, y)} \leq 1.$$

Doubling dimension. For a metric M , let $\lambda > 0$ be the smallest value such that every ball in M can be covered by λ balls of half the radius. λ is the *doubling constant* of M , and the *doubling dimension* of M is $\text{ddim}(M) = \log_2 \lambda$. Then clearly $\text{ddim}(M) = O(\log n)$. Note that while a low ℓ_p vector dimension implies a low doubling dimension – simple volume arguments demonstrate that ℓ_p metrics ($p \geq 1$) of dimension d have doubling dimension $O(d)$ – low doubling dimension is strictly more general than low ℓ_p dimension. We will often use the notation ddim when the ambient space is clear from context. The following packing property can be shown (see for example [31]):

Lemma 1.1. *Suppose that $S \subset M$ has a minimum inter-point distance α , and let $\text{diam}(S)$ be the diameter of S . Then*

$$|S| \leq \left(\frac{2 \text{diam}(S)}{\alpha} \right)^{\text{ddim}(M)}.$$

Nets and hierarchies. Given a point set S residing in metric space M , $S' \subset S$ is a γ -net of S if the minimum inter-point distance in S' is at least γ , while the distance from every point of S to its nearest neighbor in S' is less than γ . Let S have minimum inter-point distance 1. A *hierarchy* is a series of $\lceil \log \Delta \rceil$ nets (Δ being the aspect ratio of S), where each net S_i is a 2^i -net of the previous net S_{i-1} . The first (or bottom) net is $S_0 = S$, and the last (or top) net S_t contains a single point called the *root*. For two points $u \in S_i$ and $v \in S_{i-1}$, if $d(u, v) < 2^i$ then we say that u *covers* v , and this definition allows v to have multiple covering points in 2^i . The closest covering point of v is its *parent*. The distance from a point in S_i to its ancestor in S_j ($j > i$) is at most $\sum_{k=i+1}^j 2^k = 2 \cdot (2^j - 2^{i+1}) < 2 \cdot 2^j$.

Given S , a hierarchy for S can be built in time $\min\{2^{O(\text{ddim})}n, O(n^2)\}$, and this term also bounds the space needed to store the hierarchy [31, 22, 15]. (The stored hierarchy is *compressed*, in that points which do not cover any other points in the previous net may be represented implicitly.) Similarly, we can maintain links from each hierarchical point in S_i to all neighbors in net S_i within distance $c \cdot 2^i$, and this increases the space requirement to $\min\{c^{O(\text{ddim})}n, O(n^2)\}$. From a hierarchy, a *net-tree* may be extracted by placing an edge between each point $p \in S_i$ and its parent in S_{i+1} [31]. The height of the (compressed) tree is bounded by $O(\min\{n, \log \Delta\})$.

2 Background: The near neighbor problem

In this section, we review basic techniques for ANN in ℓ_p spaces. A standard technique for ANN on set $V \subset \ell_p$ is the reduction of this problem to that of solving a series of so-called approximate *near neighbor* problems [26, 21, 32] (also called the point location in equal balls problem). The c -approximate near neighbor problem for a fixed distance r is defined thus:

- If there is a point in V within distance r of query q , return some point in V within distance cr of q .
- If there is no point in V within distance r of query q , return *null* or some point in V within distance cr of q .

For example, suppose we had access to an oracle for the c -approximate near neighbor problem. If preprocess a series of oracles for the $O(\log \Delta)$ values

$$r = \{\text{diam}(V), \frac{\text{diam}(V)}{2}, \frac{\text{diam}(V)}{4}, \dots\},$$

and query them all, then clearly one of these queries would return a $2c$ -ANN of q . In particular, if r' is the distance from q to its nearest neighbor in V , then the oracle query for the value for r satisfying $r' \leq r < 2r'$ would return such a solution. Further, it suffices to seek the minimum r that returns an answer other than *null*. Then we may execute a binary search over the candidate values of r , and so $O(\log \log \Delta)$ oracle queries suffice.

Har-Peled *et al.* [21] show that for all metric spaces, the ANN problem can be solved by making only $O(\log n)$ queries to oracles for the near neighbor problem. The space requirement is $O(\log^2 n)$ times that required to store a single oracle. The reduction incurs a loss in the approximation factor, but this loss can be made arbitrarily small. In Section 4, we will require a more specialized reduction, where we allow near neighbor oracle queries only on problem instances that have constant aspect ratio.

3 ANN for ℓ_p -space via embedding into ℓ_∞

In this section, we show how to embed ℓ_p -space into ℓ_∞ in a way that guarantees that an ANN in the embedded space is also an ANN in the origin space, albeit with a degradation in the approximation quality. This allows us to extend Indyk's ℓ_∞ ANN structure to ℓ_p as well. After giving the general result in Section 3.1, we refine it in Section 3.2 to give distortion that depends on the doubling dimension of the space instead of its cardinality.

3.1 Embedding into ℓ_∞

Here we show that any n -point ℓ_p^d space admits an oblivious embedding into ℓ_∞^d with favorable properties: The embedding is non-contractive with high probability, while the interpoint expansion is small. Hence the embedding approximately preserves the nearest neighbor for a fixed query point q , and keeps more distant points far away. This implies that Indyk's ℓ_∞ ANN algorithm can be applied to all ℓ_p .

Max-stability. Our embedding follows the lead of [38, 2] in utilizing max-stable random variables, specifically those drawn from a Fréchet distribution: Having fixed p , our Fréchet random variable Z obeys for all $x > 0$

$$\Pr[Z \leq x] = e^{-x^{-p}}$$

and so

$$\Pr[Z > x] = 1 - e^{-x^{-p}} \leq x^{-p}.$$

We state the well-known max-stability property of the Fréchet distribution:

Fact 3.1. *Let random variables X, Z_1, \dots, Z_d be drawn from the above Fréchet distribution, and let $v = (v_1, \dots, v_d)$ be a non-negative valued vector. Then the random variable*

$$Y := \max_i \{v_i Z_i\}$$

is distributed as $\|v\|_p \cdot X$ (that is, $Y \sim \|v\|_p \cdot X$).

To see this, observe that

$$\begin{aligned}
\Pr[Y \leq x] &= \Pr[\max_i \{v_i Z_i\} \leq x] \\
&= \prod_i \Pr[v_i Z_i \leq x] \\
&= \prod_i \Pr[Z_i \leq x/v_i] \\
&= \prod_i e^{-(v_i/x)^p} \\
&= e^{-(\sum_i v_i^p)/x^p} \\
&= e^{-(\|v\|_p^p/x)^p}.
\end{aligned}$$

And similarly, $\Pr[\|v\|_p \cdot X \leq x] = \Pr[X \leq x/\|v\|_p] = e^{-(\|v\|_p/x)^p}$. So indeed the two random variables have the same distribution.

Embedding into ℓ_∞ . Given set $V \subset \ell_p$ of d -dimensional vectors, the embedding $f_b : V \rightarrow \ell_\infty$ (for any constant $b > 0$) is defined as follows: First, we draw d Fréchet random variables Z_1, \dots, Z_d from the above distribution. Embedding f_b maps each vector $v \in V$ to vector $f_b(v) = (bv_1 Z_1, \dots, bv_d Z_d)$. The resulting set is $V' \in \ell_\infty$. Clearly, the embedding can be computed in time $O(d)$ per point. We prove the following lemma.

Lemma 3.2. *For all $p \geq 1$, embedding f_b applied to set $V \subset \ell_p$, for $b = (3 \ln n)^{1/p}$, satisfies*

- *Contraction:* f_b is non-contractive with probability at least $1 - \frac{1}{n}$.
- *Expansion:* For any pair $u, w \in V$,

$$\|f_b(u) - f_b(w)\|_\infty \leq 2b\|u - w\|_p$$

with probability at least $1 - 2^{-p}$.

Proof. Consider some vector v with $\|v\|_p = 1$. Then by Fact 3.1, $\|f_b(v)\|_\infty \sim b\|v\|_p \cdot X = b \cdot X$, where X is a Fréchet random variable drawn from the above distribution. Then

$$\Pr[\|f_b(v)\|_\infty < 1] = \Pr[b \cdot X < 1] = e^{-(1/b)^{-p}} = \frac{1}{n^3}.$$

Since the embedding is linear, v may be taken to be an inter-point distance between two vectors in V ($v = \frac{u-w}{\|u-w\|_p}$), and so the probability that *any* inter-point distance decreases is less than $n^2 \cdot \frac{1}{n^3} = \frac{1}{n}$.

Also, $\Pr[\|f_b(v)\|_\infty > 2b] \leq 2^{-p}$, and so for any vector pair $u, w \in V$ we have $\Pr[\|f_b(u) - f_b(w)\|_\infty > 2b\|u - w\|_p] \leq 2^{-p}$. \square

Indyk's near neighbor structure is given a set $V \in \ell_\infty$ and distance r , and answers $O(\log \log d)$ -near neighbor queries for distance r in time $O(d \log n)$ and space $n^{1+\delta}$, where δ is an arbitrarily small constant (that affects the exact approximation bounds). Combining this structure and Lemma 3.2, we have:

Corollary 3.3. *Given set $V \subset \ell_p$ for $p > 2$ and a fixed distance r , there exists a data structure of size $n^{1+\delta}$ (for arbitrarily small constant δ) that solves the $O(\log \log d \log^{1/p} n)$ -approximate near neighbor problem for distance r with query time $O(d \log n)$, and is correct with probability at least $1 - \frac{1}{n} - \frac{1}{2^p}$.*

Proof. Given a set V and distance r , we preprocess the set by computing the embedding of Lemma 3.2 for each point. On the resulting set we precompute Indyk's structure for distance $r' = O(\log^{1/p} n) \cdot r$. Given a query point q , we embed the query point and query Indyk's structure. The space and runtime follows.

For correctness, by the guarantees of Lemma 3.2, if $\|q - v\|_p \leq r$ for some point $v \in V$, then under the expansion guarantee of the mapping their ℓ_∞ distance is at most $2(3 \ln n)^{1/p} \cdot r \leq r'$, so the structure does not return *null*. On the other hand, if the embedding succeeds then it is non-contractive, and so any returned point must be within ℓ_p distance $O(\log \log d) \cdot r'$ of q . The probability follows from the contraction and expansion guarantees of Lemma 3.2. \square

Finally, we use the near neighbor algorithm to solve the ANN problem, which was our ultimate goal:

Theorem 3.4. *Given set $V \subset \ell_p$ for $p > 2$, there exists a data structure of size $n^{1+\delta}$ (for arbitrarily small constant δ) which returns an $O(\log \log d \log^{1/p} n)$ -ANN in time $O(d \log^2 n) \cdot \lceil \frac{\log \log n}{p} \rceil$, and is correct with constant probability.*

Proof. We invoke the reduction of Har-Peled *et al.* [21] to reduce ANN to $O(\log n)$ near neighbor queries. We require that all $O(\log n)$ queries succeed with constant probability, hence each near neighbor query must be correct with probability $1 - O\left(\frac{1}{\log n}\right)$. Each near neighbor query is resolved by preprocessing and querying $O(\lceil \log_{2^p} \log n \rceil) = O\left(\left\lceil \frac{\log \log n}{p} \right\rceil\right)$ independent structures of Corollary 3.3. The probability that all these structures fail simultaneously is $2^{-p \cdot O((\log \log n)/p)} = O\left(\frac{1}{\log n}\right)$, and so at least one is correct with the desired probability. The runtime follows.

The reduction of [21] increases the space usage by a factor of $O(\log^2 n)$, and the additional oracles by a factor of $O\left(\left\lceil \frac{\log \log n}{p} \right\rceil\right)$, but these increases are subsumed under the constant δ in the exponent. \square

We note that when $p = \Omega(\log \log n)$, we recover the $O(\log \log d)$ -approximation guarantees of Indyk's ℓ_∞ structure, previously known to extend only to $p = \Omega(\log d)$.

3.2 Embedding with distortion dependent on the doubling dimension

Here we give an ANN algorithm whose approximation factor depends on the doubling dimension, instead of the cardinality of the space. We begin with a statement that applies only to nets. Our approach is motivated by a technique for low-dimensional Euclidean embeddings that appeared in [27].

Lemma 3.5. *Let set $V \subset \ell_p$ have minimum inter-point distance 1, and let $q \in \ell_p$ be any query point. Let $\text{ddim} \geq 2$ be the doubling dimension of $V \cup \{q\}$, and fix any value $c \geq 4$. For all $p \geq 1$, embedding f_1 (of Lemma 3.2) applied to set $V \cup \{q\}$ satisfies*

- *Contraction: Let $W \subset V$ include all points at distance at least $h = c(8 \log c \cdot \text{ddim} \ln \text{ddim})^{1/p}$ from q . Then the distance from q to W does not contract to c or less,*

$$\min_{v \in W} \|f_1(q) - f_1(v)\|_\infty > c,$$

with probability at least $1 - \text{ddim}^{-6 \text{ddim}}$.

- *Expansion:* For any pair $v, w \in V \cup \{q\}$,

$$\|f(v) - f(w)\|_\infty \leq 2\|v - w\|_p$$

with probability at least $1 - 2^{-p}$.

Proof. Let $W_i \subset W$ include all points with distance to query point q in the range $[2^i, 2^{i+1})$. Since W has minimum inter-point distance 1 and diameter less than 2^{i+1} , Lemma 1.1 implies that $|W_i| \leq 2^{(i+2) \text{ddim}}$. Let E_i be the bad event that W_i contains any point $v \in W_i$ for which $\|f_1(v) - f_1(q)\|_\infty \leq c$.

Set $j = \log h$, so that all points in W are found in sets W_{j+k} for integral $k \geq 0$. For any point $v \in W_{j+k}$ the probability that the distance from q to v contracts to c or less is

$$\begin{aligned} \Pr[\|f(v) - f(q)\|_\infty \leq c] &= e^{-(\|v-q\|_p/c)^p} \\ &\leq e^{-(2^{j+k}/c)^p} \\ &= e^{-(2^k(8 \log c \cdot \text{ddim} \ln \text{ddim})^{1/p})^p} \\ &= \text{ddim}^{-2^{k+3} \log c \cdot \text{ddim}} \\ &\leq \text{ddim}^{-2^{k+3} \log c \cdot \text{ddim}}. \end{aligned}$$

Hence, the probability of bad event E_{j+k} is at most

$$\begin{aligned} \Pr[E_{j+k}] &\leq |W_{j+k}| \cdot \text{ddim}^{-2^{k+3} \log c \cdot \text{ddim}} \\ &= 2^{(j+k+2) \text{ddim}} \cdot \text{ddim}^{-2^{k+3} \log c \cdot \text{ddim}} \\ &= 2^{(k+2+\log c) \text{ddim}} \cdot (8 \text{ddim} \ln \text{ddim})^{\text{dim}/p} \cdot \text{ddim}^{-2^{k+3} \log c \cdot \text{ddim}} \\ &\leq 2^{(k+2+\log c) \text{ddim}} \cdot (8 \text{ddim} \ln \text{ddim})^{\text{dim}} \cdot \text{ddim}^{-2^{k+3} \log c \cdot \text{ddim}} \\ &< 2^{(k+5+\log c) \text{ddim}} \cdot \text{ddim}^{(2-2^{k+3} \log c) \text{ddim}} \\ &< \text{ddim}^{(k+7+(1-2^{k+3}) \log c) \text{ddim}} \\ &\leq \text{ddim}^{(k+9-2^{k+4}) \text{ddim}} \\ &\leq \text{ddim}^{-(6+2^k) \text{ddim}} \end{aligned}$$

The probability that any point in W contracts to within distance c of q is at most $\sum_{k=0}^{\infty} \Pr[E_{j+k}] < \sum_{k=0}^{\infty} \text{ddim}^{-(6+2^k) \text{ddim}} < \text{ddim}^{-6 \text{ddim}}$, as claimed.

The expansion guarantee follows directly from Fact 3.1: $\Pr[\|f(v) - f(q)\|_\infty \geq 2\|v - q\|_p] \leq 2^{-p}$. \square

As before, Lemma 3.5 can be used to solve the near neighbor problem:

Corollary 3.6. *Given set $V \subset \ell_p$ for $p > 2$ and a distance r , there exists a data structure of size $n^{1+\delta}$ (for arbitrarily small constant δ) which solves the $O(\log \log d(\log \log \log d \text{ddim} \log \text{ddim})^{1/p})$ -approximate near neighbor problem for distance r in time $O(d \log n)$, and is correct with probability $1 - 2^{-p} - \text{ddim}^{-6 \text{ddim}}$.*

Proof. Given a set V and distance r , we preprocess the set by extracting an r -net, and then scaling down the magnitude of all vectors by r , so that the resulting set has minimum inter-point distance 1. We then compute the embedding of Lemma 3.5 into ℓ_∞ for each net-vector, and precompute Indyk's ℓ_∞ structure for distance 4. Given a query point q , we scale it down by r , embed it into

ℓ_∞ using the same embedding as before, and query Indyk's structure on distance 4. The space and runtime follows.

For correctness, let $\|q-v\|_p \leq r$ for some point $v \in V$. After extracting the net, some net-point w satisfied $\|w-v\|_p \leq r$, and so $\|q-w\|_p \leq 2r$. After scaling, we have $\|q-w\|_p \leq 2$, and after applying the embedding into ℓ_∞ , $\|q-w\|_p \leq 4$ with probability at least $1 - 2^{-p} > \frac{3}{4}$. In this case Indyk's near neighbor structure must return a point within distance $O(\log \log d)$ of q in the embedded space (that is ℓ_∞). By the guarantees of Lemma 3.5 (taking $c = O(\log \log d)$), the distance from the returned point to q in the scaled ℓ_p space is at most $O(\log \log d(\log \log \log d \text{ ddim} \log \text{ ddim})^{1/p})$, and so it is an $O(\log \log d(\log \log \log d \text{ ddim} \log \text{ ddim})^{1/p})$ -approximate near neighbor in the origin space. \square

Similar to the derivation of Theorem 3.4, we have:

Theorem 3.7. *Given set $V \subset \ell_p$ for $p > 2$, there exists a data structure of size n^{1+t} (for arbitrarily small constant t) which returns an $O(\log \log d(\log \log \log d \cdot \text{ddim} \log \text{ ddim})^{1/p})$ -ANN in time $O(d \log^2 n \log \log n) \cdot \left(\left\lceil \frac{\log \log n}{p} \right\rceil + \left\lceil \frac{\log \log n}{\text{ddim} \log \text{ ddim}} \right\rceil \right)$, and is correct with positive constant probability.*

When $p = \Omega(\log \text{ ddim} + \log \log d)$, we recover the $O(\log \log d)$ -approximation of Indyk, and this improves upon the $p = \Omega(\log \log n)$ guarantee of the previous section.

4 ANN for ℓ_p -space via embedding into ℓ_2

In this section, we show that an embedding from ℓ_p ($p > 2$) into ℓ_2 can be used to derived an $2^{O(p)}$ -ANN in logarithmic query time.

We review the guarantees of the Mazur map below, and show it can be used as an embedding into ℓ_2 . We then solve the ANN problem in the embedded space. This is however non-trivial, as the map incurs distortion that depends on the set diameter, a problem we address below.

4.1 The Mazur map

The Mazur map for the real valued vectors is defined as a mapping from L_p^m to L_q^m , $1 \leq q < p \leq \infty$. The mapping of vector $v \in L_p$ is defined as

$$M(v) = \{|v(0)|^{p/q}, |v(1)|^{p/q}, \dots, |v(m-1)|^{p/q}\},$$

where $v(i)$ is the i -th coordinate of v . The following theorem introduces a scaled Mazur map, and is adapted from [10].

Theorem 4.1. *Let $x, y \in L_p$, $p < \infty$, be vectors such that $\|x\|_p, \|y\|_p \leq C$. The Mazur map for $1 \leq q < p$ scaled down by factor $\frac{p}{q} C^{p/q-1}$ fulfills the following:*

- *Expansion: The mapping is non-expansive.*
- *Contraction: $\|M(x) - M(y)\|_q \geq \frac{q}{p} (2C)^{1-p/q} \|x - y\|_p^{p/q}$.*

The scaled Mazur map implies an embedding from ℓ_p ($p > 2$) into ℓ_2 , as in the following. (See also [33, Lemma 7.6], a significantly more general result.)

Corollary 4.2. *Any subset $V \subset \ell_p$, $p < \infty$ with $\|x\|_p \leq C$ for all $x \in V$ possesses an embedding $f : V \rightarrow \ell_2$ with the following properties for all $x, y \in V$:*

- *Expansion: The embedding f is non-expansive.*
- *Contraction: For $\|x - y\|_p = u$,*

$$\|f(x) - f(y)\|_q \geq \frac{2}{p}(2C)^{1-p/2}u^{p/2}.$$

4.2 Nearest neighbor search via the Mazur map.

Using the Mazur map, we can give an efficient algorithm for ANN for all $p > 2$. Recall that by definition, $\text{ddim} = O(\log n)$. First we define the c -bounded near neighbor problem (c -BNN) for $c > 1$ as follows: For a d -dimensional set V for which $\|x\|_p \leq c$ for all $x \in S$, given a query point q :

- If there is a point in V within distance 1 of query q , return some point in V within distance $\frac{c}{9}$ of q .
- If there is no point in V within distance 1 of query q , return *null* or some point in V within distance $\frac{c}{9}$ of q .

(The term $\frac{c}{9}$ was chosen to simplify the calculations below.)

Lemma 4.3. *For $c = p18^{p/2}$, there exists a data structure for the c -bounded near neighbor problem for $V \subset \ell_p$, $p > 2$, that preprocesses V in time and space $n^{O(1)}$, and resolves a query in time $O(d \log n)$ with probability $1 - n^{-O(1)}$.*

Proof. The points are preprocessed by first applying the scaled Mazur map to embed V into ℓ_2 in time $O(dn)$. We then use the Johnson-Lindenstrauss (JL) transform [28] (or the fast JL transform [1]) to reduce dimension to $d' = O(\log n)$ with no expansion and contraction less than $\frac{1}{2}$, in time $O(dn \log n)$. On the new space, we construct a data structure of size $2^{O(d')} = n^{O(1)}$ supporting Euclidean 2-approximate near neighbor queries in $O(d' \log n)$ time per query [21, 32].

Given a query point q , we apply the Mazur map and JL transform on the new point in time $O(d \log n)$, and use the resulting vector as a query for the 2-approximate near neighbor algorithm on the embedded space in time $O(d' \log n)$. If the point x returned by this search satisfies $\|q - x\|_p \leq \frac{c}{9}$ then we return it, and otherwise we return *null*.

To show correctness: The Mazur map is non-expansive, as is the JL transform (which is correct with probability $1 - n^{-O(1)}$). By Corollary 4.2, the Mazur map ensures that inter-point distances of $\frac{c}{9}$ or greater map to at least $\frac{2}{p}(2c)^{1-p/2}(c/9)^{p/2} = \frac{2}{p}2c18^{-p/2} = \frac{2}{p}2(p18^{p/2})18^{-p/2} = 4$, and then the contraction guarantee of the JL-transform implies that the distance in the embedded Euclidean space is greater than 2. It follows that if q possesses a neighbor in the original space at distance 1 or less, the 2-ANN in the embedded Euclidean space finds a neighbor at distance 2 in the embedded space and less than $\frac{c}{9}$ in the origin space. \square

We will show that an oracle solving c -BNN can be used as a subroutine for a data structure solving the c -approximate nearest neighbor problem. This parallels the classical reduction from ANN to the near neighbor problem utilized above. However, in order to minimize the distortion introduced by the Mazur map we must restrict the oracle to bounded diameter sets, and this results in a different reduction, adapted from [31].

Theorem 4.4. *Let V and c be as in Lemma 4.3. There exists a data structure for the $6c = 6p18^{p/2}$ -ANN problem on V , which preprocesses $\min\{2^{O(\text{ddim})}n, O(n^2)\} \cdot \left\lceil \frac{\log \log d}{p \text{ddim}} \right\rceil$ separate c -BNN oracles, each for a subset of V of size at most $w = \min\{c^{O(\text{ddim}(S))}, O(n)\}$, in total time and space $O(wnd \log \log d)$. The structure resolves a query with constant probability or correctness by executing*

$$O\left(\log d \cdot \left\lceil \frac{\log \log d}{p \text{ddim}} \right\rceil\right)$$

c -BNN oracle invocations, in total time

$$O\left(d^2 \log n + d \log d \cdot p \text{ddim} \cdot \left\lceil \frac{\log \log d}{p \text{ddim}} \right\rceil\right).$$

Proof. We preprocess a hierarchy and net-tree for V . Given query point q , we will seek a hierarchical point $w \in S_j$ which satisfies $\|w - q\|_p \leq 3 \cdot 2^j$, for minimal j . Note that if such a point w exists, then every hierarchical ancestor $w' \in S_i$ ($i > j$) of w also satisfies $\|w' - q\|_p \leq 3 \cdot 2^i$: We have $\|w' - w\|_p \leq 2 \cdot 2^i - 2 \cdot 2^j$, and so

$$\|w' - q\|_p \leq \|w' - w\|_p + \|w - q\|_p < (2 \cdot 2^i - 2 \cdot 2^j) + 3 \cdot 2^j < 3 \cdot 2^i.$$

To find w , we modify the navigating-net algorithm of Krauthgamer and Lee [31]: Beginning with the root point at the top level of the hierarchy, we descend down the levels of the hierarchy, while maintaining at each level S_i a single point of interest $t \in S_i$ satisfying $d(t, q) \leq 3 \cdot 2^i$.

Let $t \in S_i$ be the point of interest in level S_i . Then the next point of interest $t' \in S_{i-1}$ satisfies $\|t' - t\|_p \leq \|t' - q\|_p + \|q - t\|_p \leq 3 \cdot 2^i + 3 \cdot 2^{i-1} = 4.5 \cdot 2^i$. So to find t' it suffices to search all points of S_{i-1} within distance $4.5 \cdot 2^i$ of t . But there may be $2^{\Theta(\text{ddim})}$ such points, and so we cannot afford a brute-force search on the set. Instead, we utilize the c -BNN data structure of Lemma 4.3: For each net-point $t \in S_i$, preprocess a set $N(t, i)$ that includes all these candidate points of S_{i-1} , as well as all their descendants in the hierarchical level S_k , $k = i - \lceil \log 4c \rceil$ within distance $4.5 \cdot 2^i$ of t . After translating $N(t, i)$ so that t is the origin, and scaling so that all points have magnitude at most c , we preprocess for $N(t, i)$ the c -BNN oracle of Lemma 4.3. (Note that $|N(t, i)| = c^{O(\text{ddim})} = 2^{O(p \text{ddim})}$.) To find t' , execute a c -BNN query on $N(t, i)$ and q , and if the query returns a point q' , then set t' to be the ancestor of q' in S_{i-1} :

$$\|t' - q\|_p \leq \|t' - q'\|_p + \|q' - q\|_p < 2 \cdot 2^{i-1} + \frac{4.5 \cdot 2^i}{9} = 3 \cdot 2^{i-1}.$$

We terminate the algorithm in any of three events:

- The root $t \in S_i$ does not satisfy $\|q - t\|_p \leq 3 \cdot 2^i$. In this case the root itself is at worst a 3-ANN of q , as the distance from all descendants of the root (that is, all points) to q is greater than $\|q - t\|_p - 2 \cdot 2^i$.
- The algorithm locates a point $w \in S_0$ satisfying $\|w - q\| \leq 3$. Then the nearest neighbor of q can be either w or a point within distance 6 of w . We execute a c -BNN query on the set of points within distance 6 of w . (That is, as above we translate w to the origin, scale so that the maximum magnitude is c , and preprocess a query structure.) If the query returns *null*, then there is no point within distance $\frac{6}{c}$ of q , and so w is a $\frac{3}{6/c} = \frac{c}{2}$ -ANN of q . If the

query return some point, then that point is within distance $\frac{6}{9} = \frac{2}{3}$ of q . Since the minimum inter-point distance of the set is 1, there cannot be another point within distance $\frac{1}{3}$ of q , and so the returned point is a 2-ANN of q .

- A c -BNN query on some set $N(t, i)$ returns *null*. As the diameter of $N(t, i)$ is at least 2^i , this implies that there is no point in $N(t, i)$ within distance $\frac{2^i}{c}$ of q . As all descendants of a point in S_k ($k = i - \lceil \log 4c \rceil$) are within distance $2 \cdot 2^k$ of their ancestor, the distance from q to all other points is at least $\frac{2^i}{c} - 2 \cdot 2^k = \frac{2^i}{c} - 2 \frac{2^i}{4c} = \frac{2^i}{2c}$. It follows that t is a $\frac{3 \cdot 2^i}{2^i/2c} = 6c$ -ANN of q .

We conclude that the above algorithm returns the nearest neighbor stipulated by the lemma. However, its runtime depends on the number of levels in the hierarchy, that is $O(\min\{n, \Delta\})$. To remove this dependence, we first invoke the algorithm of Chan [11] to find in time $O(d^2 \log n)$ and high probability a $O(d^{3/2})$ -ANN of q , called q' . We then locate the ancestor of q' in level S_i , $i = \lceil \log \|q - q'\|_p \rceil$, of the hierarchy in time $O(\log n)$, which can be done easily using standard tree decomposition algorithms. We assign this ancestor as our first point of interest t ; note that we have

$$\|t - q\|_p \leq \|t - q'\|_p + \|q' - q\|_p < 2 \cdot 2^i + 2^i = 3 \cdot 2^i,$$

so indeed t is a valid point of interest. After descending $O(\log d)$ levels in the search, we reach radii that are smaller than the true distance from q to its nearest neighbor in V , and the search must terminate.

In order for the entire procedure to succeed with constant probability, we require the failure probability of each level c -BNN query to be $O(1/\log d)$. Each c -BNN oracle consists of $O\left(\left\lceil \frac{\log \log d}{p \text{ ddim}} \right\rceil\right)$ structures of Lemma 4.3, and (recalling that a query is executed on a set of size $2^{O(p \text{ ddim})}$) the probability that they all fail simultaneously is $2^{-O(p \text{ ddim} \cdot (\log \log d)/p \text{ ddim})} = O(1/\log d)$. The final space and runtime follow directly from the time and space required for building and querying the c -BNN oracles, each of size $\min\{O(n), 2^{O(p \text{ ddim})}\}$, plus the single level ancestor query. \square

Comment. Note that the query time is linear in the doubling dimension, as opposed to the exponential dependence common to ANN for doubling metrics. We can extend this lemma by noting that once a $2^{O(p)}$ -ANN is found, we can run the standard navigating net algorithm to descend $O(p)$ additional levels and locate a constant-factor ANN in time $p 2^{O(\text{ddim})}$. We then search $\varepsilon^{-O(\text{ddim})}$ more points in a brute-force fashion and locate a $(1 + \varepsilon)$ -ANN. So a $(1 + \varepsilon)$ -ANN can be found with $p 2^{O(\text{ddim})} + \varepsilon^{-O(\text{ddim})}$ additional work.

Acknowledgements. We thank Sarel Har-Peled, Piotr Indyk, Robi Krauthgamer, Assaf Naor and Gideon Schechtman for helpful conversations.

References

- [1] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 557–563. ACM, 2006.
- [2] A. Andoni. High frequency moments via max-stability. Manuscript, available at <http://web.mit.edu/andoni/www/papers/fkStable.pdf>, 2013.

- [3] A. Andoni, D. Croitoru, and M. Patrascu. Hardness of nearest neighbor under l -infinity. In *Foundations of Computer Science, 2008*, pages 424–433, 2008.
- [4] Sunil Arya and Theodoris Malamatos. Linear-size approximate Voronoi diagrams. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 147–155. Society for Industrial and Applied Mathematics, 2002.
- [5] K. Ball. Isometric embedding in l_p -spaces. *European J. Combin.*, 11(4):305–311, 1990.
- [6] T. Batu, F. Ergun, and C. Sahinalp. Oblivious string embeddings and edit distance approximations. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, SODA*, pages 792–801, 2006.
- [7] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.
- [8] Marshall Bern. Approximate closest-point queries in high dimensions. *Information Processing Letters*, 45(2):95 – 99, 1993.
- [9] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, pages 97–104, New York, NY, USA, 2006. ACM.
- [10] Y. Binyamini and J. Lindenstrauss. *Geometric Nonlinear Functional Analysis*. Colloquium Publications (American Mathematical Society), 2000.
- [11] T. M. Chan. Approximate nearest neighbor queries revisited. *Discrete & Computational Geometry*, 20(3), 1998.
- [12] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, September 2001.
- [13] Kenneth L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17(4), August 1988.
- [14] Kenneth L. Clarkson. Nearest neighbor queries in metric spaces. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, STOC '97*, pages 609–617, New York, NY, USA, 1997. ACM.
- [15] R. Cole and L. Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *38th annual ACM symposium on Theory of computing*, pages 574–583, 2006.
- [16] Renato Cordeiro de Amorim and Boris Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recogn.*, 45(3):1061–1075, March 2012.
- [17] András Faragó. Low distortion metric embedding into constant dimension. In *Theory and Applications of Models of Computation*, volume 6648 of *Lecture notes in computer science*, pages 114–123. Springer Berlin Heidelberg, 2011.
- [18] B. Fichet. l_p -spaces in data analysis. In H. H. Bock, editor, *Classification and Related Methods of Data Analysis*, pages 439–444. North-Holland, Amsterdam, 1988.

- [19] G.D. Finlayson, P.A.T. Rey, and E. Trezzi. General l_p constrained approach for colour constancy. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 790–797, Nov 2011.
- [20] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *The Twelfth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2004*, pages 37–41, 2004.
- [21] S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Theory of Computing*, 8(1):321–350, 2012.
- [22] S. Har-Peled and M. Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.
- [23] Sarel Har-Peled and Nirman Kumar. Approximating minimization diagrams and generalized proximity search. *SIAM Journal on Computing*, 44(4):944–974, 2015.
- [24] P. Indyk. On approximate nearest neighbors in non-euclidean spaces. In *FOCS*, pages 148–155, 1998.
- [25] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 10 – 33, 2001.
- [26] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.
- [27] P. Indyk and A. Naor. Nearest-neighbor-preserving embeddings. *ACM Trans. Algorithms*, 3(3):31, 2007.
- [28] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984.
- [29] W.B. Johnson and G. Schechtman. Embedding l_p^m into l_1^n . *Acta Mathematica*, 149(1-2):71–85, 1982.
- [30] David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, STOC '02*, pages 741–750, New York, NY, USA, 2002. ACM.
- [31] R. Krauthgamer and J.R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, January 2004.
- [32] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing, STOC '98*, pages 614–623, New York, NY, USA, 1998. ACM.
- [33] Assaf Naor. Comparison of metric spectral gaps. *Analysis and Geometry in Metric Spaces*, 2(1):1–52, 2014.

- [34] Assaf Naor and Yuval Rabani. On approximate nearest neighbor search in ℓ_p , $p > 2$. Manuscript, 2006.
- [35] Tyler Neylon. A locality-sensitive hash for real vectors. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1179–1189, 2010.
- [36] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 349–358. IEEE Computer Society, 2000.
- [37] R. Ostrovsky and Y. Rabani. Polynomial-time approximation schemes for geometric min-sum median clustering. *J. ACM*, 49(2):139–156, 2002.
- [38] S.A. Stoev, M. Hadjieleftheriou, G. Kollios, and M.S. Taqqu. Norm, point, and distance estimation over multiple signals using max-stable distributions. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1006–1015, April 2007.
- [39] Daren Yu, Xiao Yu, and Anqi Wu. Making the nearest neighbor meaningful for time series classification. In *Image and Signal Processing (CISP), 2011 4th International Congress on*, volume 5, pages 2481–2485, Oct 2011.